EE372: Quantization and Compression

Spring 2006-2007
Class Projects

April 4, 2007

As a final requirement for this class, each student is expected to do a project on data compression which uses or develops techniques, theory, or algorithms introduced in the class. In previous years, a number of class projects have resulted in conference publications, especially at the IEEE Data Compression Conference (DCC). Students are encouraged to develop a project that has some relationship to their own research interests, but a list of suggested projects is attached to help get started. Some are general and vague, some specific. Part of the creative process is finding a good project, there are no complete recipes for an acceptable project. The Web and the literature can be a big help for tracking down ideas.

Project proposals are due on Wednesday 3 May 2007 (whether or not the project is on the list), and projects themselves are due on Wednesday 6 June (by arrangement this can be extended to noon Wednesday 13 June). Oral presentations will be 4,6 June. Students are encouraged to work in pairs, but working alone is fine. (Groups of three or more are discouraged.) If two students wish to work together on a project, they must submit a joint project proposal with the individual assignments specified. All final reports should include a self contained description of the project (in readable English with appendices including any code written for the project). Each student should submit their own written report, although project partners may share "boiler plate" common material. Individual contributions should be emphasized in the reports.

A course project will typically consist of the design, programming, and simulation of one or more compression. clustering, or classification algorithms or of compression together with other signal processing. Algorithms can be applied to random process models, where mathematical performance bounds can be used for comparison, or to real data such as speech, audio, and images. Ideally algorithms should be tried on both. The reports and accompanying Web pages will be graded collectively by the TA and the instructor. The report should provide appropriate references to the literature and a comparative discussion with existing methods. The projects may be developed for any available platforms, but common languages such as C/C++, Java, Perl, and Matlab are encouraged. Projects fall within the Stanford Honor Code and all material taken from the other sources *must be completely and properly acknowledged and cited.*

The key idea is to be creative either in developing a new algorithm or in implementing an existing one in a novel way. Results whether good or bad should be compared and contrasted with the literature.

Some possibilities are listed below. You can pick a project from the list, or modify a project to suit your interests, or invent your own. Homework will be significantly reduced following the midterm to allow more time for project work.

You are encouraged to take advantage of special resources such as the scanners, super high resolution printers, etc. in the Image Systems Engineering Lab. These can be used to acquire data and display both originals and compressed reproductions.

There is code for compression system design and simulation available on the Web, e.g., Eve Riskin's C code at the University of Washington. Other resources exist on the Web and there is some local matlab code available. It is permitted to use and modify such code provided suitable acknowledgements and citations are made.

Project Tips and Guidance

1. Try to select a project that is suitable to your interests or previous experiences.

2. Once you choose your project, make a plan of how to attack the problem and apply.

3. Be aware of the all resources that you can use during your project:
    - There are several books on data compression in the Engineering Library (and possibly other libraries). See the class information sheet for a list of relevant books.
    - The *Proceedings of the IEEE Data Compression Conference.*
    - The *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing* (ICASSP)
    - The *Proceedings of the IEEE International Symposium on Image Processing* (ICIP)
    - The *IEEE Transactions on Image Processing*, as well as other related *Transactions* on *Signal Processing*, on *Speech and Audio Processing*, on *Communications*, and on *Information Theory.*
    - The usual databases, e.g., INSPEC on Folio/Socrates, Google on the Web.
    - The comp.compression group. Its FAQ sheets can be found at www.faqs.org/faqs/compression-faq/

Project Suggestions

The following are drawn from many sources, including industrial colleagues and other university research groups. With the exception of the first, they are in no particular order. The first is intended to suggest a general theme for a variety of projects.

1. Can the high-rate formulation be modified to develop results where each quantizer point in 2D space is a node in a distributed network performing some action? The idea of large numbers of nodes and densities of nodes have appeared in the networks, but little advantage has been taken of the fact that these form a spatial quantization. Martin Vetterli and others have done some work on this.

2. What are the relative merits of pruning and growing methods, especially for choosing the number of components in a Gauss mixture? Compare partition sub and supercodes with weighted codebook sub and supercodes in terms of performance and computational efficiency.

3. Locate standard databases for studying classification and try quantization-based methods and competing techniques such as the currently popular support vector machines.

4. Very little is known regarding the design of sliding-block codes, although given a sliding-decoder, the Viterbi algorithm provides an optimal encoder. Are there Lloyd-style conditions that can be exploited? Ditto for finite-state coders.

5. Distributed codes and classifiers. Effros and others have developed Lloyd-style conditions and algorithms for quantizing and classifying in networks.

6. Combined constraint optimization. Very little has been done doing Lloyd optimization with the combined entropy/codebook size constraint.

7. Convex optimization: the high-rate optimization can be formulated as a convex optimization problem. Is there an explicit solution and an optimal point density function as there is in the extreme cases of $\eta = 0, 1$?

8. “Denoising” by compression

9. New directions in quantization:
   - compressive sampling (Donoho, Candes, Nowak, Beraniuk)
   - Coarse sampling feedback quantization, generalizations of $\Sigma\Delta$ coding (Daubechies, Gunturk, Tao)
   - Iterative code design algorithms. Channel coding has undergone a revolution in recent years with near-capacity-achieving codes designed by iterative coding/message passing algorithms. Only a little work has been done trying to extend these ideas to source coding.

10. “Weakly supervised” learning. Clustering/quantization for statistical classification were image pixels are not labeled (the training images are not segmented by experts or computers). Instead each image has metadata declaring the objects they contain, but not where in the image they are. How learn the objects and identify there location?

11. Software tools: developing Matlab tools and documentation to the point where it could be made publically available and useful.

12. Global Optimality

    Some clustering algorithms for vector quantization have claimed global optimality, but only one specific result (due to Shperling, ElGamal, et al.) has ever been proved. Try to evaluate the claim by using known locally optimal algorithms with multiple starting points or by using random jumps from local optima as in simulated annealing, or by using sufficiently simple sources and small dimensions to allow exhaustive location of all local minima. The statistical technique of cross validation also provides a means towards estimating global optima. Some have claimed that deterministic annealing codes are globally optimal, but there are no proofs.

13. Clustering for discrete sources

    Clustering algorithms such as Lloyd have long been known to perform poorly on discrete sources with the average Hamming distortion. The problem is an enormous number of poor local optima. With todays computing power you could find true global optima for simple problems by exhaustion, e.g., for iid binary sources with fixed rate and modest block size. Can you find any properties of the true optima or even good local optima that would assist a clustering algorithm in avoiding poor local optima, i.e., additional optimality properties for the Lloyd algorithm past the usual nearest neighbor, centroid, and zero-probability boundary rules. One problem here is to efficiently index and test the codes and to not retest codes that are essentially the identical to previously tested codes.

14. Classified lossy compression

    Improved compression can often be obtained by using different codes locally in an image to code distinct types, such as background, text, graphics, and photos. Explore what has been done and consider a variation or two. Particular attention should be paid to simplicity of implementation. What shapes of segmented regions are allowed has a strong impact on complexity. Both classified and universal vector quantization can be used.

15. Quality in lossy compression

    (a) Implement and compare different proposed perceptual distortion measures for image compression. Design vector quantizers to optimize with respect to these measures. Design an experiment to compare the quality of the resulting images both subjectively and objectively. How well do the computable perceptual distortion measures predict human subjective distortion?

    (b) Perceptual model development for compression: Look up methods that have been used to incorporate perceptual models (HVS) into compression algorithms and develop and test a variation. E.g., apply DCT ideas developed at Bell (Safranek and Johnson) or Ames (Watson and Ahumada) to wavelet transforms. (There has been quite a bit of work on this, some of it pretty bad.)

16. Dithering has an interesting practical and theoretical history. Two approaches to the mathematics (and history) can be found in papers by Gray and Stockham (*IEEE Trans. Inform. Thy*) and Wannamaker et al. (IEEE Trans. Signal Processing). Some of the results have been extended to lattice quantizers (e.g., Feder, Zamir). Can similar results be found for nonuniform quantizers? Attempt to analyze Gaussian dither. (See Bennett's classical derivation of quantization noise for Gaussian processes and Davenport and Root's derivation of the spectral density of hard limited Gaussian processes.)

    A particular issue: the theory is all for genuine dither, that is, genuinely random processes added to the signal and possibly removed from the reconstructed signal. In practice, however, a pseudo-random noise signal is used instead. Although intuitively reasonable, the theory appears to break down entirely. Is there a theory for pseudorandom dither?

17. Minimum Discrimination Length (MDL) for continuous data. Rissanen, Barron, Yu and others developed the MDL principle for modeling data. The classical derivation was for discrete sources and was based on Shannon noiseless coding ideas. The original extension of MDL to continuous data assumed uniform quantization. Develop an extension using optimal quantization and compare the approaches.